LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-641635

Non-Galerkin Coarse Grids for Algebraic Multigrid

R. D. Falgout, J. B. Schroder

July 31, 2013

SIAM Journal of Scientific Computing

Disclaimer

Non-Galerkin Coarse Grids for Algebraic Multigrid

Robert D. Falgout[†] and Jacob B. Schroder[†]

August 1, 2013

Abstract

Algebraic multigrid (AMG) is a popular and effective solver for systems of linear equations that arise from discretized partial differential equations. While AMG has been effectively implemented on large scale parallel machines, challenges remain, especially when moving to exascale. In particular, as problem size increases, so does the number of levels in the hierarchy of coarse grids with the pernicious result that coarse grid operators tend to become denser further down in the hierarchy. This produces denser communication patterns than existed on fine grids and as a result, as problem size increases, denser coarse grid matrices are produced, and the communication pattern and overall parallel AMG scheme become less efficient. This increase in density is due to the standard Galerkin coarse grid operator, P^TAP, where P is the prolongation (i.e., interpolation) operator. Typical choices for P couple most distance two and many distance three connections in the graph of A, and this results in the increasing fill. For example, the coarse grid stencil size for a simple 3D 7-point finite differencing approximation to diffusion can increase into the thousands on present day machines, causing an associated increase in communication costs. We therefore consider algebraically truncating coarse grid stencils to obtain a non-Galerkin coarse grid. First, the sparsity pattern of the non-Galerkin coarse grid is determined through the fine grid matrix graph, which indicates important distance two and three connections that should be maintained on the coarse grid. Second, the nonzero entries are determined by collapsing the stencils in the Galerkin operator using traditional AMG techniques. The result is a reduction in coarse grid stencil size, overall operator complexity, and parallel AMG solve phase times.

1 Introduction

The goal of this paper is to introduce a non-Galerkin coarse grid strategy that improves the parallel performance of algebraic multigrid (AMG) [4, 12]. AMG is of interest because it is a popular and effective solver of large sparse linear systems in parallel, by virtue of being *scalable* or *optimal* and solving a linear system with N unknowns with only $O(N)$ work. Consider solving the linear system

$$A\mathbf{x} = \mathbf{b}, \tag{1}$$

where $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$. For the problems considered here, $A \in \mathbb{R}^{n\times n}$ is assumed to be a symmetric and positive definite (SPD) matrix.

There are two main components to a multigrid method: the *smoother* (or *relaxation* method) and the *coarse grid correction* step. The coarse grid correction step involves operators that transfer information between the fine and coarse "grids", denoted more generally by the space $\mathbb{R}^n$ and the lower dimensional (coarse) vector space $\mathbb{R}^{n_c}$. We focus on classical Ruge-Stüben style AMG (RS-AMG) [12], which constructs the coarse space using a disjoint F/C splitting of the fine grid unknowns, with the C-points forming the coarse

[†]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551, email: schroder2@llnl.gov, falgout2@llnl.gov. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-JRNL-641635)

grid. We let $P : \mathbb{R}^{n_c} \to \mathbb{R}^n$ be the *interpolation* (or *prolongation*) operator, and generally the *restriction* operator is taken to be $R = P^T$. Optimality is achieved when smoothing and coarse grid correction are complementary. In the classical setting of scalar elliptic problems, this means that the smoother is a simple iterative algorithm like Gauss-Seidel, which is effective at reducing high frequency error. The remaining low frequency error is then accurately represented and efficiently eliminated on coarser grids via the coarse grid correction step. The result, is that all error frequencies are uniformly damped with a linear cost. The two-grid multigrid method for solving (1) is then defined as follows:

$$\text{Relax } \nu_1 \text{ times on } A\mathbf{x} = \mathbf{b}. \tag{2a}$$

$$\text{Correct } \mathbf{x} \leftarrow \mathbf{x} + P(RAP)^{-1}R(\mathbf{b} - A\mathbf{x}). \tag{2b}$$

$$\text{Relax } \nu_2 \text{ times on } A\mathbf{x} = \mathbf{b}. \tag{2c}$$

In practice, we use a multilevel method to solve (1) by recursively applying algorithm (2) to the correction step in (2b). The proposed work considers replacing the Galerkin coarse-grid operator RAP with a sparser approximation A_c, which maintains the AMG convergence rate, but improves parallel efficiency.

Controlling the sparsity pattern of RAP is important because AMG (including the target BoomerAMG package in *hypre*) produces coarse grids with a maximum coarse grid stencil size that grows with problem size. This is problematic in parallel, because as density increases on coarser levels, standard matrix partitioning couples processors that were not coupled on previous, finer levels. Thus, the communication pattern also becomes denser as problem size increases, with the risk that for exascale problems, the maximum coarse grid stencil size may produce unscalable communication patterns. This increase in communication is well documented [14, 6]. In particular, the work [6] describes how even the best current practices in *hypre* still result in a sharp increase in number of messages and in overall communication cost when moving to coarse grids. For example, the model diffusion problem considered in [6] resulted in a growth from 6 MPI sends on the finest level to 245 MPI sends on the worst coarse grid for a machine with 65K cores. The result was that the time spent on some coarse levels in the hierarchy, despite being much smaller in terms of number of nonzeros and number of unknowns, was actually larger than the time spent on the finest level.

The need for reductions in coarse grid sparsity has been a main driver for research-based improvements to *hypre*. Classical parallel AMG (i.e., classical modified interpolation and Falgout coarsening [7]) is indeed a useful and powerful method, but at large numbers of cores, the communication cost ruins what is otherwise a computationally optimal method. As an example, consider Figure 1, where the time to solution for classical parallel AMG is plotted for the simple 3D 7-point finite difference diffusion operator. A weak scaling study is done with 25,000 unknowns per core on an Intel cluster with a fast Infiniband QDR interconnect, and the time grows dramatically with core count. This result led to the important developments of aggressive coarsening [14] and extended interpolation [13], which lead in this experiment to a reduction in coarse grid stencil size from the thousands to the hundreds. The resulting time to solution is also commensurately reduced, in large part due to reduced communication, as indicated by the "Best Practices AMG" plot in Figure 1. However, even with these advancements, the time to solution is still growing in Figure 1, leaving us room for improvement through communication reducing strategies.

Our strategy is to further reduce the communication costs by replacing RAP with a sparser approximation. While we focus on a purely algebraic approach, non-Galerkin methods have already been explored in settings where geometric information is used to aid the method [18, 1] in choosing sparsity patterns and matrix coefficients. This previous work already indicates that a significant reduction in coarse grid stencil size is possible without a serious reduction in multigrid convergence. We note that the work [3] discusses a purely algebraic non-Galerkin coarse grid approach, but requires a prohibitively large computational cost. More recent work [15] has followed a similar path to ours in sparsifying coarse grid operators, and while not fully developed or tested in parallel, shows much promise.

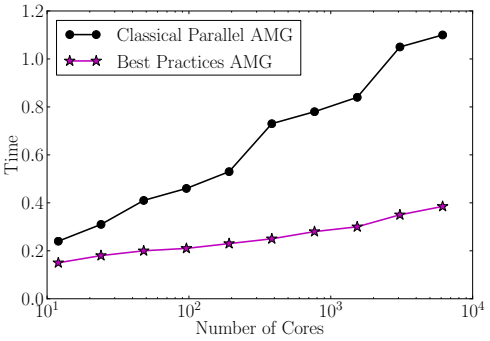


Figure 1: AMG weak scaling as motivation: parallel timings for classical parallel AMG and current best practices for a simple model diffusion problem

Our overall strategy is to begin with an existing AMG method, and to remove entries from the Galerkin operator each time a new level in the hierarchy is formed. Thus our goal is to have sparser coarse grid operators than given by Galerkin coarse grids and to also maintain AMG convergence when compared to the corresponding Galerkin coarse grid method. The algorithm consists of two phases. The first phase chooses a sparsity pattern for the new coarse grid operator, and the second phase removes or "collapses" entries in P^TAP that lie outside of that sparsity pattern. While we focus on the *hypre* package, the proposed method has also been briefly and successfully tried in the smoothed aggregation (SA) [17] setting, and we believe that the approach is general enough to be applicable to most parallel AMG codes.

In Section 2, we first provide a mathematical motivation for our approach. In Section 3, we state the resulting algorithm. In Section 4, we describe a near optimal non-Galerkin coarse grid operator, and compare our method to the near optimal method for model diffusion problems. In Section 5, we give serial results for a variety of diffusion and elasticity problems, followed by parallel results for diffusion problems.

2 Mathematical Motivation

In this section, we provide the mathematical motivation for our approach. Assume that an AMG method is applied to an SPD matrix, A. During construction of an AMG hierarchy, the Galerkin coarse grid, $A_g = RAP$, has been computed, but the rows contain too many nonzeros, or the number of nonzeros per row is growing too quickly. Thus, we seek to replace A_g with a sparser approximation, A_c, that is spectrally equivalent, i.e.,

$$\forall \mathbf{x} \in \mathbb{R}^n,\ \exists \alpha, \beta \in \mathbb{R}^+ \text{ such that} \quad \alpha \leq \left\langle A_c^{-1} A_g \mathbf{x},\, \mathbf{x} \right\rangle \leq \beta, \tag{3}$$

with α and β both close to 1. However, we show that minimizing

$$\left\| I - A_c A_g^{-1} \right\|_2 \tag{4}$$

is a sufficient heuristic to both target spectral equivalence and to minimize the eventual multigrid convergence factor.

We begin by restating previous results [9], that assume A_c is SPD. Define the two-grid error propagator for AMG as

$$E_{TG} = (I - M^{-T}A)^\nu (I - PK^{-1}P^TA)(I - M^{-1}A)^\nu, \tag{5}$$

where K is the coarse grid matrix and M defines the relaxation method, e.g., the diagonal of A in the case of Jacobi. If $K = A_g$, then equation (5) is the standard Galerkin two-grid error

propagator with symmetric pre- and post-relaxation. The asymptotic AMG convergence factor is the spectral radius $\rho(E_{TG})$, and is our starting point for developing the target A_c.

Next, we denote the preconditioner B_{TG} for A induced by (5) with $K = A_g$ as

$$E_{TG} = I - B_{TG}^{-1}A. \tag{6a}$$

If $K = A_c$, then this perturbed coarse grid operator yields

$$E_{PTG} = I - B_{PTG}^{-1}A. \tag{6b}$$

The associated convergence factors are then

$$\rho(E_{TG}) = \max(\lambda_{\max}(B_{TG}^{-1}A) - 1, \;\; 1 - \lambda_{\min}(B_{TG}^{-1}A)) \tag{7a}$$
$$\rho(E_{PTG}) = \max(\lambda_{\max}(B_{PTG}^{-1}A) - 1, 1 - \lambda_{\min}(B_{PTG}^{-1}A)), \tag{7b}$$

where we use the fact that A, B_{PTG} and B_{TG} are SPD and that the eigenvalues of $(B_{PTG}^{-1}A)$ and $(B_{TG}^{-1}A)$ are therefore all real and positive.

Theorem 2.1. *Define E_{PTG} and B_{TG} as above and let*

$$\theta = \left\| I - A_c A_g^{-1} \right\|_2. \tag{8}$$

If $\theta < 1$ and both A_g and A_c are SPD, then

$$\rho(E_{PTG}) \leq \max\left(\lambda_{\max}(B_{TG}^{-1}A) \cdot \frac{1}{1-\theta} - 1, \; 1 - \lambda_{\min}(B_{TG}^{-1}A) \cdot \frac{1}{1+\theta} \right). \tag{9}$$

Proof. Importantly, the perturbed two-grid convergence factor (7b) is bounded in terms of the Galerkin approach (7a) and an error term in [9], yielding

$$\lambda_{\max}(B_{PTG}^{-1}A) \leq \lambda_{\max}(B_{TG}^{-1}A) \;\cdot\; \max(\lambda_{\max}(A_c^{-1}A_g), 1), \tag{10a}$$
$$\lambda_{\min}(B_{PTG}^{-1}A) \geq \lambda_{\min}(B_{TG}^{-1}A) \;\cdot\; \min(\lambda_{\min}(A_c^{-1}A_g), 1). \tag{10b}$$

Overall, we desire to maximize $\min(\lambda_{\min}(A_c^{-1}A_g), 1)$ from (10b), but simultaneously minimize $\max(\lambda_{\max}(A_c^{-1}A_g), 1)$ from (10a). We assume that $B_{TG}^{-1}A$ is fixed.

Now, we move the inverse from A_c to A_g, which will be convenient later when we specify the sparsity pattern of A_c. Since A_g and A_c are SPD, then $\lambda(A_c^{-1}A_g) > 0$ and $\lambda(A_g^{-1}A_c) > 0$, and we have

$$\max(\lambda_{\max}(A_c^{-1}A_g), 1) = \max\left(\frac{1}{\lambda_{\min}(A_g^{-1}A_c)}, 1 \right) \tag{11a}$$
$$= \frac{1}{\min(\lambda_{\min}(A_g^{-1}A_c), 1)} \equiv \frac{1}{\mu_{\min}}, \tag{11b}$$

and analogously,

$$\min(\lambda_{\min}(A_c^{-1}A_g), 1) = \frac{1}{\max(\lambda_{\max}(A_g^{-1}A_c), 1)} \equiv \frac{1}{\mu_{\max}}. \tag{11c}$$

So, we want to maximize $\mu_{\min}$ while simultaneously minimizing $\mu_{\max}$. This is achievable by considering the connection to the 2-norm and using the assumption that A_g and A_c are SPD,

$$\rho(I - A_g^{-1}A_c) = \max\left(\lambda_{\max}(A_g^{-1}A_c) - 1, 1 - \lambda_{\min}(A_g^{-1}A_c) \right), \tag{12a}$$
$$= \max\left(\mu_{\max} - 1, 1 - \mu_{\min} \right) \tag{12b}$$
$$\leq \left\| I - A_g^{-1}A_c \right\|_2 = \left\| I - A_c A_g^{-1} \right\|_2. \tag{12c}$$

The eventual algorithm focuses on minimizing (12a), which first targets $\lambda_{\max} < 2$, and then minimizes $\lambda_{\max}$ and $\lambda_{\min}$ with equal weight. With $\mu_{\max} - 1$, and $1 - \mu_{\min}$ both bounded by $\theta < 1$, we can go back to (10a) and (10b) to yield

$$\lambda_{\max}(B_{PTG}^{-1}A) \leq \lambda_{\max}(B_{TG}^{-1}A) \cdot \frac{1}{1-\theta}, \tag{13a}$$

$$\lambda_{\min}(B_{PTG}^{-1}A) \geq \lambda_{\min}(B_{TG}^{-1}A) \cdot \frac{1}{1+\theta}, \tag{13b}$$

This leads (by substituting into (7b)) to the final statement, which yields a convergence rate bound that deteriorates with respect to E_{TG} as θ increases. □

Equation (9) is the goal, because it expresses the final two-grid convergence rate in terms of the convergence rate of the standard Galerkin coarse grid and θ a term that measures spectral equivalence between A_c and A_g. The algorithm will seek to make θ close to 0.

2.1 Heuristics

In this section, we further motivate the algorithm heuristics based on Theorem 2.1. Our basic goal is to construct an operator A_c with a given sparsity pattern such that θ is small. To state this more rigorously, we first introduce some notation. Define a *matrix nonzero pattern* (or *sparsity pattern*) as a set of tuples $\{(i,j)\}$ and denote the space of matrices with given sparsity pattern $\mathcal{N}$ by

$$\boldsymbol{\mathcal{N}} \equiv \{A \in \mathbb{R}^{n\times n} : a_{ij} \neq 0 \text{ only if } (i,j) \in \mathcal{N}\}. \tag{14}$$

Similarly, define a *vector nonzero pattern* as a set of indices $\{i\}$ and denote the space of vectors with given nonzero pattern $\mathcal{I}$ by

$$\boldsymbol{\mathcal{I}} \equiv \{\mathbf{x} \in \mathbb{R}^{n} : x_j \neq 0 \text{ only if } j \in \mathcal{I}\}. \tag{15}$$

Let $\mathcal{N}_c$ and $\mathcal{N}_g$ represent the sparsity patterns of A_c and A_g, respectively. Then, our aim is to define $\mathcal{N}_c \subseteq \mathcal{N}_g$ and $A_c \in \boldsymbol{\mathcal{N}}_c$ such that θ is small. We begin by stating A_c as a perturbation of A_g, i.e., let $A_c \in \boldsymbol{\mathcal{N}}_c, A_g \in \boldsymbol{\mathcal{N}}_g, E \in \boldsymbol{\mathcal{N}}_g$, and

$$A_c = A_g + E, \tag{16a}$$

$$\left\|I - A_c A_g^{-1}\right\|_2 = \left\|E A_g^{-1}\right\|_2 = \theta. \tag{16b}$$

Next, we develop heuristics for (16b) intended to reduce θ. First, we develop a heuristic for "mid-range" or high energy modes, $\mathbf{x}$,

$$A_g\mathbf{x} = \lambda\mathbf{x}, \text{ such that } \|\mathbf{x}\|_2 = 1. \tag{17a}$$

Our simple row-wise heuristic uses Gershgorin rings, i.e., we enforce

$$\|\mathbf{e}_i\|_1 \leq \gamma \left\|\mathbf{a}_i^g\right\|_1, \tag{17b}$$

where $(\mathbf{e}_i)^T$ and $(\mathbf{a}_i^g)^T$ represent the ith row and $\gamma \in [0,1]$. Thus letting $\max_i \left\|\mathbf{a}_i^g\right\|_1 = k\rho(A_g)$, we have

$$\rho(E) \leq \gamma \max_i \left\|\mathbf{a}_i^g\right\|_1 \tag{17c}$$

$$\rho(E) \leq k\gamma\, \rho(A_g) \tag{17d}$$

Last for the mode $\mathbf{x}$ in question, we can say

$$\left\|E A_g^{-1}\mathbf{x}\right\|_2 = \frac{1}{\lambda}\|E\mathbf{x}\|_2 \leq \frac{\rho(E)}{\lambda} \leq k\gamma\frac{\rho(A_g)}{\lambda} \tag{17e}$$

For typical AMG problems from standard discretizations of the Poisson operator, k is small, usually in the vicinity of 2 or 3. Thus for mid-range or high energy modes, say when $\lambda \in [\rho(A_g)/10,\, \rho(A_g)]$, the impact on θ from equation (17e) is minimal for the γ chosen here, which are on the order 0.01.

Much of the preceding argumentation for mid-range to high-energy modes centers around global eigenvalues, but it is important that the heuristic (17b) is row wise. This locality implies that enforcing (17b) for a specific γ targets the locally mid-range to high energy modes, which may or may not correspond to the global mid-range to high energy modes.

The other case concerns $\mathbf{x}$ as a low energy eigenmode such that $\|\mathbf{x}\|_2 = 1$, but $A_g\mathbf{x} \approx \mathbf{0}$. Then, θ is small when

$$E\mathbf{x} \approx \mathbf{0} \quad \text{if} \quad A_g\mathbf{x} \approx \mathbf{0}. \tag{18a}$$

Since, AMG typically makes a priori assumptions about the local behavior of the near null space, we can leverage that here as well to target (18a) as a heuristic. In particular, we enforce accuracy for a set of vectors, B, that represent the near null space, i.e.,

$$A_cB = A_gB \quad \Leftrightarrow \quad EB = \mathbf{0}. \tag{18b}$$

The standard choice for B is the constant, but could also be problem dependent, e.g., the rigid-body modes for elasticity. For the most common case of $B = \mathbf{1}$, we guarantee (18b) with classical AMG inspired stencil collapsing to remove the unwanted entries in A_g, so that $E\mathbf{1} = \mathbf{0}$. For other cases, a more sophisticated projection based strategy is used, which we detail later.

While this approach is based on the above heuristics, it should be noted that much of the successful classical AMG framework, e.g., strength-of-connection and interpolation formulas, is based on similar heuristic assumptions. Put another way, if the constant does not adequately represent a low energy vector $\mathbf{x}$ locally, then the classical AMG framework in which we operate will already be problematic.

3 Algorithm

In this section, we first describe our algorithm for finding a suitable sparsity pattern $\mathcal{N}_c$ for A_c. Second, we describe the algorithm for eliminating entries in A_g based on $\mathcal{N}_c$ for the case of classical AMG scalar problems. The strategy is based on classical AMG stencil collapsing. Third, we discuss one possible generalization of our algorithm to other PDEs, such as linearized elasticity, where the near null space of the matrix contains more than the constant vector.

The heuristics in Section 2 guide our algorithmic choices. The process of finding the sparsity pattern utilizes a drop tolerance that guarantees satisfaction of equation (17b), which is the heuristic targeting accuracy in A_c for mid-range to high energy modes. The process of stencil collapsing to eliminate entries is then done in such a way to target the second heuristic (18b) and give A_c accuracy for the important near null space modes.

3.1 Choosing the Sparsity Pattern

The goal of the sparsity pattern choice is to ensure an adequate nonzero pattern for spectral equivalence between A_g and A_c. We approach this problem in a two phase manner. The first phase initializes the sparsity pattern using the matrix graph of P and the fine grid operator A to target sufficient connections in A_c to approximate the Galerkin matrix stencil. The second phase guaranties heuristic (17b) to more directly target spectral equivalence for mid-range to high energy modes.

Minimal Sparsity Pattern

Considering the first phase, we must at a minimum choose coarse grid sparsity patterns that *allow* for reproducing a good approximation to the Galerkin matrix stencil. We choose

$$\hat{\mathcal{N}}_c = \{(i,j) \text{ such that } (P_I^T A P + P^T A P_I)_{ij} \neq 0\} \tag{19}$$

as the initial sparsity pattern, where P_I refers to injection between the coarse and fine grids. This initial pattern allows for important matrix entries in A_g with a small magnitude to be preserved on coarse grids in $\hat{\mathcal{N}}_c$. These entries would normally be dropped if the classical strength-of-connection measure, which is based on magnitude, were used to compute $\hat{\mathcal{N}}_c$. Instead equation (19) uses the information (often of a geometric nature) present in the matrix graph of A to maintain connections on the coarse grid between all fine grid distance one C-point connections and some fine grid distance two C-point connections. (This assumes that classical RS-AMG type interpolation and coarsening schemes are used.) So while we will later use a collapsing scheme based on RS-AMG interpolation to eliminate entries in A_g, the concept of strength-of-connection needed to construct $\mathcal{N}_c$ is different than for RS-AMG. We note that [15] uses a similar approach in a SA setting to obtain a sparsity pattern.

To illustrate this concept, consider a simple example of grid aligned anisotropic diffusion, $-u_{xx} - \epsilon u_{yy} = f$, discretized with the standard five-point finite-differencing stencil, depicted by A in equation (20a). The stencil for A_g in (20a) is that yielded by one level of semi-coarsening with RS-AMG. The stencil for A_c in (20b) is that obtained from rediscretizing the problem on the semi-coarsened grid (i.e., no coarsening in the y-direction, while choosing every other unknown in the x-direction). The matrix A_c provides a suitable non-Galerkin coarse grid for A_g by virtue of being spectrally equivalent to A_g. This then yields the fact that $\|I - A_c^{-1} A_g\|$ is bounded independent of h, or in terms of equation (8), that $\|I - A_c A_g^{-1}\|$ is bounded independent of h. However if the connections in the direction of weak diffusion are dropped in A_c to yield $\hat{A}_c$, the bound for $\|I - \hat{A}_c{}^{-1} A_g\|$ is no longer h-independent, and Theorem 2.1 indicates that using $\hat{A}_c$ as the non-Galerkin coarse grid will not guarantee an h-independent two-grid method.

$$A := \begin{bmatrix} & -\epsilon & \\ -1 & (2+2\epsilon) & -1 \\ & -\epsilon & \end{bmatrix}, \qquad A_g := \begin{bmatrix} -\frac{\epsilon}{4} & -\frac{3}{2}\epsilon & -\frac{\epsilon}{4} \\ -\frac{1}{2}+\frac{\epsilon}{2} & (1+3\epsilon) & -\frac{1}{2}+\frac{\epsilon}{2} \\ -\frac{\epsilon}{4} & -\frac{3}{2}\epsilon & -\frac{\epsilon}{4} \end{bmatrix}, \tag{20a}$$

$$A_c := \begin{bmatrix} & -2\epsilon & \\ -\frac{1}{2} & (1+4\epsilon) & -\frac{1}{2} \\ & -2\epsilon & \end{bmatrix}, \qquad \hat{A}_c := \begin{bmatrix} & 0 & \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ & 0 & \end{bmatrix}, \tag{20b}$$

To see this h-dependence, we do a simple derivation using the well known [5] eigenvectors v_i and eigenvalues λ_i of the classic second-order finite-difference operator with stencil $\begin{bmatrix} -1 & 2 & -1 \end{bmatrix}$. Letting this 1D stencil be defined on a grid with n points, the corresponding spectrum of A_c defined on an $n \times n$ grid is

$$\lambda_{ij} = \frac{1}{2}\lambda_i + 2\epsilon\lambda_j, \tag{21a}$$

where the indexing $i, j < n$. The eigenvectors of A_c are

$$v_{ij} = v_i \otimes v_j. \tag{21b}$$

The spectrum of $\hat{A}_c$ can similarly be expressed with

$$\hat{A}_c v_{ij} = \frac{1}{2}\lambda_i v_{ij}. \tag{21c}$$

Using this knowledge of the spectra,

$$\left\| I - \hat{A}_c{}^{-1} A_c \right\|_2 = \lambda_{max} \left(\left(I - \hat{A}_c{}^{-1} A_c \right)^T \left(I - \hat{A}_c{}^{-1} A_c \right) \right)^{1/2} \tag{21d}$$

$$= \max_{ij} \left| 1 - \frac{\frac{1}{2}\lambda_i + 2\epsilon\lambda_j}{\frac{1}{2}\lambda_i} \right| \tag{21e}$$

$$= \max_{ij} \; 4\epsilon\lambda_i^{-1}\lambda_j \tag{21f}$$

$$\lesssim 4\epsilon \left(\frac{4-h}{h} \right). \tag{21g}$$

In summary, h-independence for θ is achieved when the closest small magnitude entries in the direction of weak diffusion are in the stencil for A_c. We avoid the h-dependent sparsity pattern of $\hat{A}_c$ by using $\hat{\mathcal{N}}_c$ from (19) as the minimal sparsity pattern, which for this example is

$$\hat{\mathcal{N}}_c := \begin{bmatrix} & * & \\ * & * & * \\ & * & \end{bmatrix}. \tag{22}$$

If only the classical concept of strength-of-connection were employed to determine $\hat{\mathcal{N}}_c$ based on A_g, then the pattern for $\hat{A}_c$ would be chosen. It is also important that the pattern $\hat{\mathcal{N}}_c$ does not include the longer distance diagonal connections in A_g of small magnitude, which are not necessary for a spectrally equivalent coarse grid. In fact, it is the elimination of such entries that allows for stencil size reductions in our experiments.

Improving the Sparsity Pattern

Next, we improve $\hat{\mathcal{N}}_c$ by enforcing the heuristic (17b) through the following procedure, in order to target accuracy for mid-range to high energy modes. Let the set of neighbors of i in $\mathcal{N}_c$ (i.e., the allowed nonzeros in row i) be

$$\mathcal{N}_{ci} = \{j, \text{ such that the tuple } (i,j) \in \mathcal{N}_c\}.$$

The procedure then begins by initializing the pattern $\mathcal{N}_c$ with the pattern of A_g. Entries are then removed from $\mathcal{N}_c$, starting with the smallest in magnitude, until any further removal of entries would violate

$$2 \sum_{j \notin \mathcal{N}_{ci}} |a_{ij}^g| \leq \gamma \sum_j |a_{ij}^g|. \tag{23}$$

Assuming classical stencil collapsing will be done to compute final matrix entry values (as outlined in Section 3.2), this procedure guarantees (17b) for the final A_c, even before the stencil collapsing step. The factor of 2 essentially represents the change made to A_g when dropping an entry, plus the maximum change to A_g possible when collapsing that entry to the allowed nonzero entries. Last, we ensure the minimal sparsity pattern from (19), by taking the union of $\mathcal{N}_c$ and $\hat{\mathcal{N}}_c$, i.e., we replace $\mathcal{N}_c$ with $\mathcal{N}_c \bigcup \hat{\mathcal{N}}_c$. Algorithm 1 gives a detailed description of this process. If a symmetric pattern is desired, $\mathcal{N}_c$ can be symmetrized, such that if $(i,j) \in \mathcal{N}_c$, then (j,i) is added to the set $\mathcal{N}_c$ as well.

3.2 Eliminating Entries in A_g for Scalar Problems

The goal when eliminating entries in A_g to obtain A_c is to not change the action on the near null space of A_g. We rely on the sparsity pattern choice from Section 3.1 to avoid changing the action on other modes. We eliminate entries in A_g with a classical AMG stencil collapsing based approach, because this approach has been used for decades [4, 12, 16] to develop interpolation formulas which are accurate for the constant based near null space modes

Algorithm 1: `ComputeSparsity`(A_g, P, P_I)

Input: A_g {Galerkin Operator}
P {Interpolation operator}
P_I {Injection operator}

$\mathcal{N}_c \leftarrow \emptyset$
for (i,j) *such that* $a^g_{ij} \neq 0$ **do**
  $\mathcal{N}_c \leftarrow \mathcal{N}_c \bigcup \{(i,j)\}$ {Initial pattern}

for $i = 1$ **to** $nrows(A_g)$ **do**
  Initialize set $\mathcal{K}$: $\mathcal{K}_m$ is index of mth smallest off-diagonal nonzero in row i
  for $m = 1$ **to** $|\mathcal{K}|$ **do**
    $\mathcal{N}_{ci} \leftarrow \mathcal{N}_{ci} \setminus \mathcal{K}_m$ {Tentatively remove entry $\mathcal{K}_m$ }
    if $2\sum_{j \notin \mathcal{N}_{ci}} |a^g_{ij}| \leq \gamma \sum_j |a^g_{ij}|$ **then**
      continue {Equation (17b) Satisfied }
    else
      $\mathcal{N}_{ci} \leftarrow \mathcal{N}_{ci} \bigcup \mathcal{K}_m$ {Equation (17b) Violated}
      break

for (i,j) *such that* $(P_I^T AP + P^T AP_I)_{ij} \neq 0$ **do**
  $\mathcal{N}_c \leftarrow \mathcal{N}_c \bigcup \{(i,j)\}$ {Union with minimal pattern}
return $\mathcal{N}_c$

critical for classical scalar problems. In other words, this lumping procedure is (roughly speaking) as effective at preserving the near null space as classical AMG interpolation is accurate for the near null space.

The dropping strategy begins by initializing A_c as a copy of A_g. Then, each $a^c_{ij} \neq 0$ such that $(i,j) \notin \mathcal{N}_c$ is eliminated from A_c. The stencil collapsing strategy is to eliminate a^c_{ij} by adding a fraction of that entry to each of j's strongly connected neighbors in row i. The fractional lumping is done such that the row sum does not change and the local constant based near null space is thus unchanged. We begin by letting S be the strength-of-connection matrix defined by the classical AMG measure with respect to A_g, and define the neighbors of j in S as

$$\mathcal{N}_{sj} = \{k, \text{ such that } s_{jk} \neq 0\}. \tag{24a}$$

Next, we find U, which represents the strong connections of j shared by the nonzero pattern of row i. This set U represents the unknowns to which a^c_{ij} will be lumped.

$$U = \mathcal{N}_{sj} \bigcap \mathcal{N}_{ci}. \tag{24b}$$

If strong neighbors are found ($U \neq \emptyset$), we avoid changing the diagonal and remove i from U. This avoids changing the center of each row's Gershgorin disc, if possible. We then compute the fractions by which the entry will be lumped, and carry out the fractional lumping such that the near null space is preserved and $E\mathbf{1} = 0$:

$$\sigma = \sum_{k \in U} |s_{jk}| \tag{24c}$$

$$a^c_{ik} \leftarrow a^c_{ik} + (|s_{jk}|/\sigma)\, a^c_{ij} \quad \text{for } k \in U. \tag{24d}$$

Similar to classical stencil collapsing, if no strong neighbors are found ($U = \emptyset$), the entry is lumped to the diagonal. After all entries are eliminated, the matrix A_c may be symmetrized,

$$A_c \leftarrow 0.5(A_c^T + A_c), \tag{24e}$$

followed by diagonal lumping to preserve row sum,

$$a_{ii}^c \leftarrow a_{ii}^c + \sum_j a_{ij}^g - \sum_j a_{ij}^c. \tag{24f}$$

Algorithm 2 describes in detail this stencil collapsing algorithm.

Algorithm 2: `NonGalerkin`$(A_g, S, \mathcal{N}_c)$

```
1  Input: A_g                                          {Galerkin Operator}
2         S                          {Strength-of-connection operator based on A_g}
3         N_c                                         {Desired sparsity pattern}
4
5  A_c ← A_g                                  {Initialize with Galerkin coarse grid}
6  for i = 1 to nrows(A_c) do
7      for j such that a_ij ≠ 0 do
8          if j ∉ N_ci then
9              U ← N_sj ⋂ N_ci
10             if U = ∅ then
11                 a_ii^c ← a_ii^c + a_ij^c                {No neighbors, lump to diagonal}
12             else
13                 U ← U \ {i}                                  {Avoid changing diagonal}
14                 σ = Σ_{k∈U} |s_jk|
15                 for k ∈ U do
16                     a_ik^c ← a_ik^c + (|s_jk|/σ) a_ij^c              {Lump to neighbors}
17             a_ij^c ← 0
18         else
19             Do nothing
20 A_c ← 0.5(A_c^T + A_c)                                                {Symmetrize}
21 for i = 1 to nrows(A_c) do
22     a_ii^c ← a_ii^c + Σ_j a_ij^g − Σ_j a_ij^c                        {Preserve row sum}
23 return A_c
```

While the stencil collapsing process preserves the row sum of the operator, the preservation of symmetry in line 20 changes this row sum. Thus, line 22 is required to modify the diagonal and preserve the row sum. The values added to the diagonal in line 22 are typically small, but necessary for maintaining accuracy for the near null space of the operator. From a different point of view, this symmetrization step can be thought of as a way of symmetrically dropping entries, where lines 15 and 16 are replaced with

$$\textbf{for } k \in U \textbf{ do} \quad a_{ik}^c \leftarrow a_{ik}^c + (1/2)(|s_{jk}|/\sigma)a_{ij}^c \tag{25a}$$

$$a_{ki}^c \leftarrow a_{ki}^c + (1/2)(|s_{jk}|/\sigma)a_{ij}^c, \tag{25b}$$

followed by a fix up step that lumps to the diagonal to preserve row sum.

Remark 1. *The stencil collapsing approach of Algorithms 1 and 2 does preserve definiteness for M-matrices. An M-matrix is a common multigrid operator where the diagonal is positive, all off-diagonals are non-positive, and row sums are non-negative. Thus by a Gershgorin disc argument, M-matrices are at least positive semi-definite. When the proposed approach*

operates on an M-matrix, all off-diagonals remain non-positive (off-diagonals are changed only by adding a negative value); the diagonal remains positive; and the row sum is unchanged. Thus the resulting A_c is still an M-matrix and hence definite. If the symmetrization step is done, then A_c is SPD. If the interpolation method chosen preserves the M-matrix property on coare grids, then the proposed method will still yield an SPD preconditioner. However for general matrices, there is no way to guarantee that the proposed method preserves definiteness. This necessitates the use of more general Krylov methods such as GMRES or BiCGStab, when using AMG as a preconditioner.

In summary, this algorithm removes entries from a coarse grid operator in a way that preserves the spectrum for constant like near null space modes. The algorithm relies on the choice of $\mathcal{N}_c$ to guarantee that the resulting coarse grid is accurate for mid-range to high energy modes. Additionally, the algorithm relies primarily on local row wise computations that require little communication, making the method well suited for the parallel setting.

3.3 Eliminating Entries in A_g in a General Setting

In this section, we consider the case where the matrix requires stencil collapsing that preserves multiple modes in B, i.e., we must enforce heuristic (18b) when B has multiple columns. The motivation is to extend the proposed method to problems such as elasticity, where the near null space consists of multiple vectors, e.g., the rigid-body modes. We first present classical stencil collapsing in terms of the general approach, and then discuss one way to extend this perspective to the multiple mode case.

Consider Figure 2, where the matrix graph relative only to unknown 1 is depicted, and a local ordering is used. In the picture, unknowns 5 and 6 are to be eliminated from the stencil, and collapsed to unknowns 2, 3 and 4. The gray dotted lines and associated weights represent strength-of-connection couplings between 5 and 6 relative to their neighbors 2, 3 and 4. The strength-of-connection couplings have been normalized such that the two connections for unknown 5 sum to 1, and likewise for unknown 6.

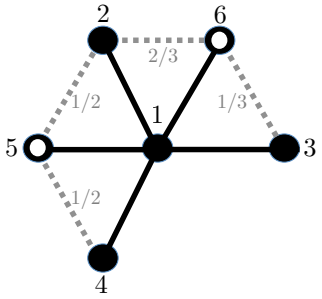


Figure 2: Stencil Collapsing Example

Let the set of off-diagonal unknowns that are maintained in row $i = 1$ be denoted by

$$\mathcal{N}_{ci} = \{2, 3, 4\}, \tag{26}$$

and the set of off-diagonal unknowns to be eliminated be denoted by

$$\chi_i = \mathcal{N}_{gi} \setminus \mathcal{N}_{ci} = \{5, 6\}, \tag{27}$$

Then the corresponding generalization for classical stencil collapsing can be represented with a local interpolation like operator G_i, that operates only on nonzero entries, i.e.,

$$\begin{bmatrix} a_{ii}^c & (\mathbf{a}_i^c|_{\mathcal{N}_{ci}})^T \end{bmatrix} = \begin{bmatrix} a_{ii}^g & (\mathbf{a}_i^g|_{\mathcal{N}_{ci}})^T & (\mathbf{a}_i^g|_{\chi_i})^T \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & I \\ 0 & G_i \end{bmatrix} \tag{28a}$$

where $\mathbf{a}_i^c|_{\mathcal{N}_{ci}}$ is the vector of off-diagonal entries after stencil collapsing, $a_{ii}^c = a_{ii}^g$, and

$$G_i = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 2/3 & 1/3 & 0 \end{bmatrix}. \tag{28b}$$

Note that all other entries in row i of A_c are zero. The entry $(G_i)_{0k}$ represents the normalized strength-of-connection entry between unknowns 5 and k, and $(G_i)_{1k}$ likewise corresponds to unknown 6. More precisely, recall the definition of $U \leftarrow \mathcal{N}_{sj} \bigcap \mathcal{N}_{ci}$ for each row i from Algorithm 2, and define G_i as

$$G_i \in \mathbb{R}^{|\chi_i| \times |\mathcal{N}_{ci}|} \tag{29a}$$

$$(G_i)_{uv} = \frac{s_{(\chi_i)_u,(\mathcal{N}_{ci})_v}}{\sum_{k \in U} |s_{(\chi_i)_u,k}|} \tag{29b}$$

In summary, the operator G_i interpolates from unknowns in $\mathcal{N}_{ci}$ to unknowns in χ_i using normalized strength-of-connection values that interpolate between strongly connected unknowns, and preserve near null space mode(s). For this example, all row sums of G_i equal 1. In general, classical stencil collapsing will always result in a row sum of 1 in G_i.

Remark 2. *Equation (28a) defines the general case of stencil collapsing, but it does not allow for entries to be lumped to the diagonal as in the Algorithm 2. This can be observed because the first column is all zero, except for the (1,1) entry. This was an engineering choice meant to protect the diagonal entry from large changes during the process of fitting multiple modes into the span of G_i, which can cause relatively large changes to entries.*

3.3.1 Accommodating Multiple Near Null Space Modes

We now present the generalized stencil collapsing case, where G_i is post-processed to accommodate multiple near null space modes. This set of vectors B is usually specified on the finest level, and then transferred to each coarse level with the restriction operator $P_I^T B$. Multigrid methods typically require a set of such modes for construction of interpolation. Even classical AMG methods implicitly assume $B = \mathbf{1}$ by virtue of the interpolation formulas which are explicitly designed to preserve the constant.

Our strategy updates the G_i from classical stencil collapsing in a least-squares fashion so that G_i interpolates multiple vectors B accurately. The intention is that the minimal update results in a G_i with weights that still favor strong connections. More precisely, we update G_i in a minimal 2-norm sense to fit B and satisfy

$$G_i B|_{\mathcal{N}_{ci}} = B|_{\chi_i}, \tag{30a}$$

where

$$B|_{\mathcal{N}_{ci}} = B_{uv} \quad \text{for } u \in \mathcal{N}_{ci} \text{ and } v \in \{0, 1, 2, ..., ncols(B)\} \tag{30b}$$

$$B|_{\chi_i} = B_{uv} \quad \text{for } u \in \chi \text{ and } v \in \{0, 1, 2, ..., ncols(B)\}. \tag{30c}$$

This property guaranties that the action of $A_c B = A_g B$. To see how (30a) accomplishes this, take equation (28a) and right multiply by

$$\begin{bmatrix} \mathbf{b}_i^T \\ B|_{\mathcal{N}_{ci}} \end{bmatrix}, \tag{31}$$

using the identity (30a). The quantity $\mathbf{b}_i^T$ represents B restricted to the ith row. We now describe in Algorithm 3 the generalized approach to stencil collapsing.

In addition to Remark 2, there are some important algorithmic notes to make. When solving PDE systems, such as elasticity, where an unknown j belongs to only one of many

Algorithm 3: `Generalized_NonGalerkin`$(A_g, S, \mathcal{N}_c, B)$

Input: A_g {`Galerkin Operator`}
S {`Strength-of-connection operator based on` A_g}
$\mathcal{N}_c$ {`Desired sparsity pattern`}
B {`Near null spacemodes for` A_g`, ordered in importance`}

$A_c \leftarrow A_g$ {`Initialize with Galerkin coarse grid`}
for $i = 1$ **to** $nrows(A_c)$ **do**
  $k \leftarrow \max\left(\left\lfloor \frac{|\mathcal{N}_{ci}|}{2} \right\rfloor - 2, 1\right)$
  Compute G_i {`Equation (29)`}
  $F \leftarrow B|\chi_i - G_i B|\mathcal{N}_{ci}$ {`Compute update to` G_i}
  for $j = 1$ **to** $nrows(G_i)$ **do**
    Let $\mathbf{f}$ and $\mathbf{g}$ be row j of F and G_i, respectively.
    $\mathbf{u} \leftarrow \text{argmin}_{\mathbf{u}} \|(B|\mathcal{N}_{ci})^T\mathbf{u}^T - \mathbf{f}^T\|_2^2$
    $\mathbf{g} \leftarrow \mathbf{g} + \mathbf{u}$
  $\begin{bmatrix} a_{ii}^c & (\mathbf{a}_i^c|_{\mathcal{N}_{ci}})^T \end{bmatrix} = \begin{bmatrix} a_{ii}^g & (\mathbf{a}_i^g|_{\mathcal{N}_{ci}})^T & (\mathbf{a}_i^g|_{\chi_i})^T \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & I \\ 0 & G_i \end{bmatrix}$ {`Equation (28a) `}
  $\mathbf{a}_i^c|_{\chi_i} \leftarrow 0$ {`Zero out entries not in pattern`}
return A_c

PDEs, it is critical to limit the collapsing to occur between unknowns in the same PDE. This can easily be implemented in Algorithm 3 by pre-filtering the strength matrix S with

$$s_{ij} \leftarrow 0 \quad \text{if} \quad \text{PDE\_num}(i) \neq \text{PDE\_num}(j), \tag{32}$$

where the function PDE_num() returns the PDE number (say 0, 1 or 2) for an unknown. In order to compute the PDE number for a coarse unknown, we utilize the C/F splitting present at each level to find the corresponding fine grid unknown for each coarse grid unknown.

Remark 3. *We have observed issues with over fitting and the computation of G_i. As an example, let G_i have 4 nonzeros per row, and B contain 4 or 5 vectors, then the process of computing G_i to exactly preserve all modes is ill conditioned, i.e., small changes to B can result in large changes to G_i. We have observed this phenomenon for the model 2D rotated anisotropic diffusion problems in the results section, when B equals the constant, linears and quadratics. Here, the coefficients computed for G_i would sometimes vary by 2 or 3 orders of magnitude. The resulting coarse grid stencil became inaccurate, and the overall AMG cycle deteriorated or diverged as a result. As a solution, we take only the first k_i columns of B when computing G_i. It is assumed that B is ordered in importance, so that the first k_i vectors are the most critical ones. We choose k_i to be significantly smaller than the number of nonzeros per row in G_i, or $k = |\mathcal{N}_{ci}|/2$, where $|\mathcal{N}_{ci}|$ is the number of unknowns in $\mathcal{N}_{ci}$. Thus, the number of modes in B used to compute G_i can change with i. While this strategy is purely an engineering choice, it is a reasonable number of modes to choose from B to avoid over fitting.*

4 Evaluating the Accuracy of the Stencil Collapsing Approach

In this section, we describe a near optimal non-Galerkin coarse grid operator, and compare it to our method for model diffusion problems. Let $(\mathbf{a}_i^c)^T$ be the ith row of A_c and let λ_j and $\mathbf{v}_j$ be the eigenvalues and eigenvectors of A_g. Below, we derive an upper bound for θ in (8)

in terms of $\mathbf{a}_i^c$, λ_j and $\mathbf{v}_j$. In (33d), we write $\mathbf{x} = \sum_j \alpha_j \mathbf{v}_j$, with $\alpha_j \in [0,1]$ and $\sum_j |\alpha_j| = 1$. In (33e), we can reverse the order of the min-max problem because it is quadratic in α_j and a_{ij}^c. A useful upper bound on the minimal θ is then

$$\begin{aligned}
\theta_{\min}^2 &= \min_{A_c \in \boldsymbol{\mathcal{N}}_c} \left\| (I - A_c A_g^{-1}) \right\|_2^2 && \text{(33a)} \\
&= \min_{A_c \in \boldsymbol{\mathcal{N}}_c} \max_{\|\mathbf{x}\|_2 = 1} \left\| (I - A_c A_g^{-1})\mathbf{x} \right\|_2^2 && \text{(33b)} \\
&= \min_{A_c \in \boldsymbol{\mathcal{N}}_c} \max_{\|\mathbf{x}\|_2 = 1} \sum_i ((\mathbf{x})_i - (\mathbf{a}_i^c)^T A_g^{-1}\mathbf{x})^2 && \text{(33c)} \\
&= \min_{A_c \in \boldsymbol{\mathcal{N}}_c} \max_{\|\boldsymbol{\alpha}\|_2 = 1} \sum_i \left(\sum_j \alpha_j (v_{ij} - \lambda_j^{-1} \mathbf{v}_j^T \mathbf{a}_i^c) \right)^2 && \text{(33d)} \\
&= \max_{\|\boldsymbol{\alpha}\|_2 = 1} \min_{A_c \in \boldsymbol{\mathcal{N}}_c} \sum_i \left(\sum_j \alpha_j (v_{ij} - \lambda_j^{-1} \mathbf{v}_j^T \mathbf{a}_i^c) \right)^2 && \text{(33e)} \\
&= \max_{\|\boldsymbol{\alpha}\|_2 = 1} \sum_i \min_{\mathbf{a}_i^c \in \boldsymbol{\mathcal{N}}_{ci}} \left(\sum_j \alpha_j (v_{ij} - \lambda_j^{-1} \mathbf{v}_j^T \mathbf{a}_i^c) \right)^2 && \text{(33f)} \\
&\leq \sum_i \min_{\mathbf{a}_i^c \in \boldsymbol{\mathcal{N}}_{ci}} \sum_j \left(v_{ij} - \lambda_j^{-1} \mathbf{v}_j^T \mathbf{a}_i^c \right)^2 . && \text{(33g)}
\end{aligned}$$

To compute our near optimal A_c, we note that the $\mathbf{a}_i^c$ which satisfy equation (33g) and thus ultimately bound $\theta_{\min}$ are equivalent to

$$\operatorname*{argmin}_{\mathbf{a}_i^c \in \mathcal{N}_{ci}} \left\| \begin{bmatrix} \mathbf{v}_{i1} \\ \mathbf{v}_{i2} \\ \vdots \\ \mathbf{v}_{in} \end{bmatrix} - \begin{bmatrix} (1/\lambda_1)(\mathbf{v}_1^T)|_{\mathcal{N}_{ci}} \\ (1/\lambda_2)(\mathbf{v}_2^T)|_{\mathcal{N}_{ci}} \\ \vdots \\ (1/\lambda_n)(\mathbf{v}_n^T)|_{\mathcal{N}_{ci}} \end{bmatrix} (\mathbf{a}_i^c)|_{\mathcal{N}_{ci}} \right\|_2^2 . \tag{34}$$

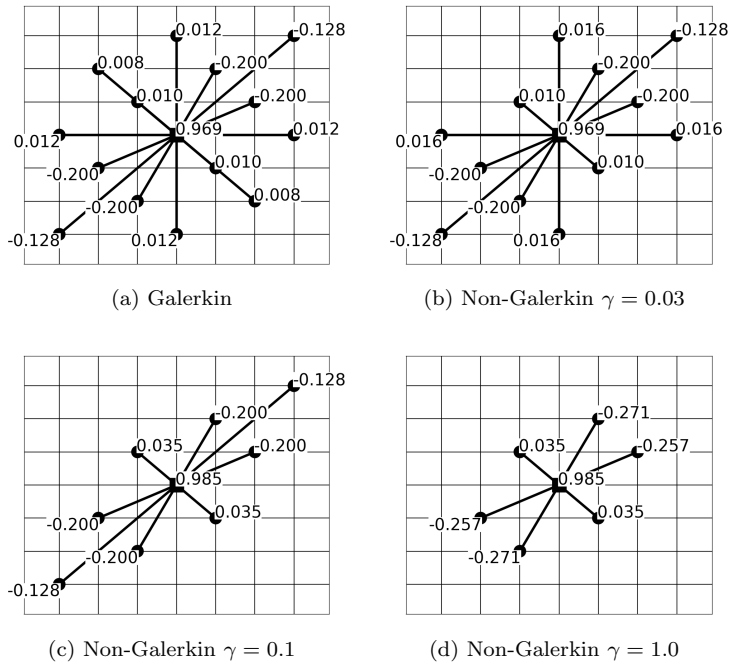
Hence, the near optimal A_c is simply constructed by solving (34) for each row i.

To use this tool, we compared our method to the near optimal A_c on a variety of model 2D diffusion problems which yielded small enough matrices that the computation of the near optimal A_c was possible. We asked two questions. One, is the two-level AMG convergence rate significantly better with the near optimal method when compared to the proposed approach? Two, what sparsity pattern is necessary to achieve equivalent performance to the Galerkin approach when using the near optimal method?

As an example, consider the Q1 bilinear finite element discretization on a regular structured grid with Dirichlet boundaries of the rotated anisotropic diffusion equation

$$-(c^2 + \epsilon s^2)u_{xx} - 2(1-\epsilon)cs\, u_{xy} - (\epsilon c^2 + s^2)u_{yy} = f, \tag{35}$$

where $\epsilon = 0.001$, $c = \cos(\theta)$, $s = \sin(\theta)$ and θ is the angle of rotation. We choose the angle $\theta = \pi/4$ (which is a particularly difficult angle for the stencil collapsing approach, on a 15×15 grid). The stencil at the center point on the first coarse grid is given in Figures 3a, 3b, 3c and 3d for the cases of the Galerkin approach, and for the proposed stencil collapsing approach with $\gamma = 0.03$, $\gamma = 0.1$ and $\gamma = 1.0$. The asymptotic convergence rates for Figures 3a, 3b, 3c and 3d are 0.23, 0.23, 0.40 and 0.47, respectively. The corresponding convergence rates and stencils for the near optimal approach are nearly identical and are hence omitted. The one exception was for $\gamma = 1.0$, where the near optimal approach did yield a better convergence rate and slightly different coefficient values in the cross-stream direction. Thus, the conclusion is two-fold. One, that the longer distance connections present in Figure 3b, but not present in Figures 3c and 3d, are necessary to regain the performance of the Galerkin



(a) Galerkin

(b) Non-Galerkin $\gamma = 0.03$

(c) Non-Galerkin $\gamma = 0.1$

(d) Non-Galerkin $\gamma = 1.0$

Figure 3: Center point stencils for first coarse grid operators, comparing Galerkin and stencil collapsing based non-Galerkin for various γ.

approach when using either the near optimal or stencil collapsing approach. Two, the coefficients computed by stencil collapsing yield similar performance to that achieved by fitting the eigenvectors in the near optimal approach.

A number of different angles and finite difference approximations for isotropic and anisotropic examples were explored with similar results. So, this tool led us to conclude that in general a broader sparsity pattern than $R_I AP$ was needed and that $\gamma = 0.03$ yielded such a sparsity pattern for the diffusion problems examined. This tool also allowed us to conclude that the coefficients computed by stencil collapsing are as good as the near optimal approach for smaller values of γ.

5 Results

In this section, we examine the numerical performance of the method for a variety of diffusion test problems and two 3D elasticity problems. In general, the tests use a V-cycle of RS-AMG to accelerate GMRES with a relative residual tolerance of 10^{-8}. The tests also maintain identical solver parameters and only toggle the non-Galerkin method on and off, so that the standard Galerkin approach and the new method can be fairly compared. The serial tests were done using the PyAMG [2] package, while the parallel tests were done using the *hypre* [8] package.

5.1 Scalar Diffusion Results

In this section, we examine the proposed non-Galerkin method for scalar diffusion problems. V(2,2) cycles of classical RS-AMG using weighted-Jacobi are used to accelerate GMRES with a relative residual tolerance of 10^{-8}. No post-processing symmetrization of the coarse grid operator is used, because it did not noticeably change convergence. For all of the test problems, 4 consecutive refinements are considered for increasing matrix sizes of roughly

16 000, 64 000, 256 000 and 1 000 000. This is true, even for the 3D problems.

The tables of results contain the following columns. "Max Stencil" reports the maximum stencil size over all levels, and is used as a proxy for the amount of communication induced by the AMG hierarchy. The term ρ_{AMG} refers to the convergence rate. "Op. Comp." refers to operator complexity, which is the total number of nonzeros stored in the hierarchy, divided by the number of finest level nonzeros. "Work" refers to work per digit of accuracy, i.e., the amount of work, in matrix-vector multiplies, required to reduce the residual by one order of magnitude. Work is particularly useful when comparing methods with differing operator complexities. A variety of γ values are experimented with, including the case of $\gamma = 1.0$, which is equivalent to using $\hat{\mathcal{N}}_c$ as the sparsity pattern, and $\gamma = 0.0$, which is equivalent to using the Galerkin coarse grid approach. The overall goal of the tests is to reduce stencil size and hence parallel communication while only negligibly affecting AMG convergence.

In 2D, we consider the rotated anisotropic diffusion equation (35). The discretizations considered are standard second-order finite differencing and Q1 bilinear finite elements on regular structured grids with Dirichlet boundaries. Experiments are run for a variety of θ values, with worst case angles occurring for roughly $\theta \approx 2\pi/16$, which is the test case we choose to show in Table 1. While this is a worst case, it is still representative of other angles, in that as γ decreases to $[0.01, 0.03]$, the proposed approach attains similar convergence and nearly identical work units to the Galerkin approach ($\gamma = 0.0$). Overall, the non-Galerkin approach is robust for these problems, and provides a reduction in operator complexity and stencil size.

γ	refinement	ρ_{AMG}	Op. Comp.	Work	Max stencil
1.0	0	0.28	2.3	20	16
	1	0.44	2.3	25	16
	2	0.51	2.3	32	16
	3	0.56	2.3	36	17
0.03	0	0.31	2.3	17	16
	1	0.39	2.3	23	16
	2	0.46	2.3	28	16
	3	0.50	2.3	31	16
0.01	0	0.28	2.5	19	21
	1	0.31	2.6	21	22
	2	0.35	2.6	23	21
	3	0.37	2.6	24	22
0.0	0	0.26	2.7	19	33
	1	0.28	2.7	20	33
	2	0.30	2.8	21	33
	3	0.32	2.8	23	33

Table 1: 2D non-Galerkin results for various γ, rotated anisotropic diffusion by angle of $\theta = 2\pi/16$, AMG preconditioning GMRES.

The next set of experiments examine 3D diffusion operators. We examine the problem

$$u_{xx} + \epsilon_y u_{yy} + \epsilon_z u_{zz} = f \tag{36}$$

using classic second-order 7-point finite differencing approximations on regular grids and also linear tetrahedral discretizations on unstructured isotropic meshes. The PDE coefficients considered are $\epsilon_y = 1.0$, $\epsilon_z = 0.001$ and $\epsilon_y = 1.0$, $\epsilon_z = 1.0$.

The results are broadly similar, so we give a representative set in Table 2 for the linear tetrahedral example and isotropic coefficients ($\epsilon_y = 1.0$, $\epsilon_z = 1.0$). The non-Galerkin approach is robust, preserves the Galerkin AMG convergence rate and yields a stencil size

reduction of 50%. One important note is that the operator complexity reduction is not always as dramatic as depicted in Table 2. For instance for the 7-point finite-differencing Poisson operator, the operator complexity reduction is only from 2.9 to 2.8, but the non-Galerkin approach still enjoys a roughly 50% stencil size reduction.

γ	refinement	ρ_{AMG}	Op. Comp.	Work	Max stencil
0.03	0	0.18	1.5	8	52
	1	0.24	1.6	10	76
	2	0.30	1.8	14	81
	3	0.36	1.9	17	81
0.0	0	0.18	1.6	8	73
	1	0.24	1.8	11	91
	2	0.30	2.2	17	150
	3	0.38	2.6	24	174

Table 2: 3D non-Galerkin results for two γ choices, isotropic diffusion, AMG preconditioning GMRES.

5.2 Elasticity Results

We next examine the proposed approach for isotropic linearized elasticity problems, where $B \in \mathbb{R}^{n,6}$ and is equal to the six rigid body modes. Linearized elasticity is defined by

$$-\text{div}\left(\lambda \text{ tr}\left(\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)/2\right) I + \mu\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)\right) = f, \tag{37}$$

where λ and μ are the Lamé parameters, I is the identity matrix and tr() is the trace function. The GetFem++ package [11] is used to discretize. The test problems examined include an iron bar attached to the left wall with a downward force applied to the top of the bar. The iron bar is defined on the region $[0,4] \times [0,1] \times [0,1]$ and discretized with parallelpiped elements. The other test problem is a steel tripod corresponding to the mesh in Figure 4. Here, the tripod is discretized using linear tetrahedral elements and a downward external force is applied to the top of the tripod.

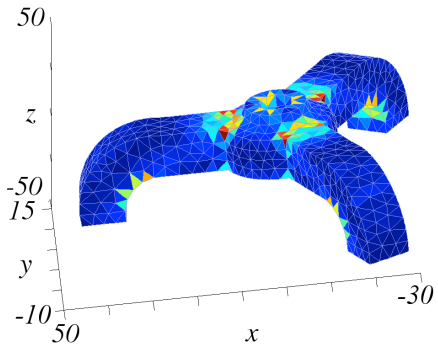


Figure 4: Example tetrahedral tripod mesh.

The solver setup is different than in Section 5.1, because the interpolation operators for RS-AMG only capture the constant vector. Hence, the experiments with RS-AMG show no performance difference when comparing Algorithm 2 with Algorithm 3 and B equal the six rigid-body modes. Essentially, if $P^T AP$ does not accurately represent the rigid-body modes, then the stencil collapsing approach does not need to account for them. So while the non-Galerkin RS-AMG approach applied to the elasticity test problems is effective,

we consider a smoothed aggregation (SA) approach [10] instead, that allows span(P) to capture all six of the rigid-body modes, and hence $P^T AP$ accurately represents all six of the rigid-body modes. We also choose a SA approach to show the generality of the proposed non-Galerkin method to most AMG methods.

While SA generally does not associate C-points with the coarse grid, this set of unknowns can be obtained by associating a root-node [10] with each aggregate. This root node is usually the aggregate seed during the greedy aggregation process [17]. These seeds then form the C-point sets and corresponding P_I operators.

The convergence results are given in Table 3 for the tripod example, with the results for the iron bar being similar, and therefore omitted. Three consecutive refinements are considered for increasing matrix sizes of roughly 3 000, 17 000 and 109 000. The second column denotes the values for B chosen, with "RBM" standing for the six rigid-body modes, and $B = \mathbf{1}$ representing the standard stencil collapsing of Algorithm 2. The choice of $B = \mathbf{1}$ is similar to collapsing with the three displacement rigid-body modes, which likely explains why Algorithm 2 performs well.

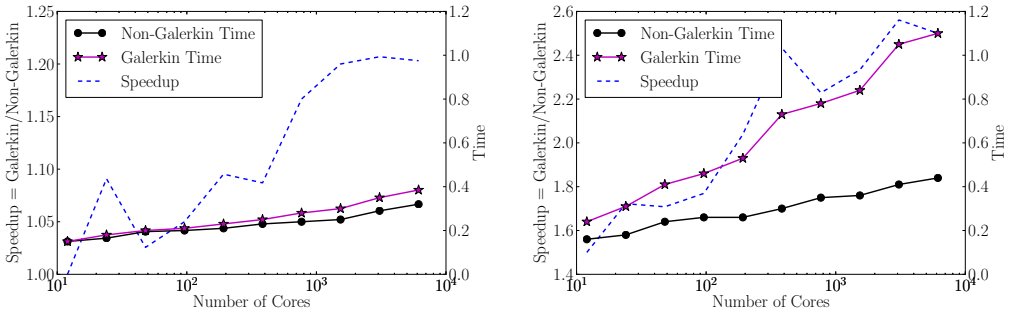
The proposed approach matches the convergence rate of the Galerkin approach for $\gamma = 0.03$, with a significant drop in maximum stencil size. Interesting, for smaller values of γ, such as $\gamma = 0.03$, the results are nearly identical for B equalling the six rigid-body modes or the constant. For the very restricted sparsity patterns of $\gamma = 1.0$, collapsing with respect to the rigid body modes becomes important, and even allows for regaining the original work per digit of accuracy numbers for the Galerkin approach.

γ	B	refinement	ρ_{AMG}	Op. Comp.	Work	Max stencil
1.0	$\mathbf{1}$	0	0.45	1.2	12	42
		1	0.50	1.2	16	72
		2	0.64	1.2	24	97
1.0	RBM	0	0.44	1.2	12	42
		1	0.48	1.2	14	72
		2	0.54	1.2	18	97
0.03	$\mathbf{1}$ or RBM	0	0.40	1.3	11	56
		1	0.44	1.4	15	138
		2	0.48	1.5	18	245
0.0	***	0	0.39	1.4	11	69
		1	0.43	1.5	16	182
		2	0.47	1.6	20	346

Table 3: Elasticity non-Galerkin results for various γ and collapsing strategies, AMG preconditioning GMRES.

5.3 Parallel Results

In this section, we present parallel results for the simple scalar 7-point 3D isotropic diffusion finite difference operator with Dirichlet boundary conditions. The goal is to demonstrate the potential of the algorithm to speedup parallel AMG solve phase times. We therefore choose the simple 7-point model problem that nonetheless yields an unscalable growth in stencil size. The parallel stencil collapsing is identical to the serial algorithm, with a few exceptions. The collapsing is done with a binary strength-of-connection matrix (i.e., each strong connection has a value of 1.0 and each weak connection has a value of 0.0). This is done because binary strength-of-connection matrices is what is currently available in *hypre*. Implementing the complete stencil collapsing approach from Algorithm 2 using standard strength matrices is future work.



(a) Best Current Parallel AMG (b) Classical Parallel AMG

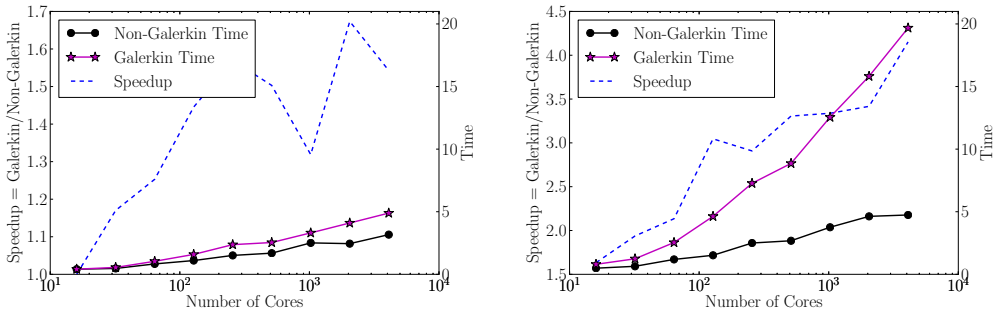
Figure 5: Speedup and AMG solve phase timings comparing non-Galerkin AMG and Galerkin AMG on machine with *fast* network.

The parallel studies carried out are weak scaling tests with 25,000 unknowns per core on two different machines, one with a fast InfiniBand QDR interconnect and the other with a slow InfiniBand DDR interconnect. V(1,1) cycles using hybrid Gauss-Seidel are used to precondition GMRES to within a relative residual tolerance of 10^{-8}. We also explore two different AMG scenarios, one that uses a "best practices" set of parameters (e.g. aggressive coarsening [14] and extended interpolation [13]) and the other that uses classical parallel AMG parameters (e.g. classical modified interpolation and Falgout coarsening [7]). We report speedup and time to solution for the AMG solve phase in order to compare the Galerkin and non-Galerkin approaches. The stencil size reductions are similar to that reported in serial, and are therefore omitted here. Yet, it is important to note that these stencil size reductions are what translate into the reduced solve phase times.

Figures 5a and 5b show the speedup and timings for the best practices scenario and classical parallel AMG scenario, respectively, on the machine with a fast interconnect. The speedup is on the left axis for the dotted line, and the time to solution for the AMG solve phase is on the right axis for the two solid lines depicting the Galerkin and non-Galerkin approaches. For the best practices scenario, there is a speedup of 15–20%, but for the classical parallel AMG scenario, the speedup grows to 250%. Figures 6a and 6b depict the similar results for the machine with a slow interconnect, but with more pronounced speedups of 150% and 400% for the best practices and classical parallel AMG scenarios, respectively.

These results are encouraging, in part because the speedup grows with number of cores. Thus, we expect this approach to yield even better speedups when moving to current Petascale machines (10^6 cores) and also to next generation exascale machines (10^6–10^9 cores).

Moreover, the results raise the question whether the current best practices in AMG can be avoided altogether, in favor of a more classical AMG approach that leverages non-Galerkin coarse grids. For instance, if the overall time is compared between the best practices and classical parallel AMG scenarios for each machine, it is observed that the non-Galerkin approach for the classical scenario is competitive with the Galerkin approach for the best practices scenario. This rethinking of parallel AMG is attractive because the current best practices for reducing stencil size and communication (i.e., aggressive coarsening and extended interpolation) also worsen the AMG convergence rate, whereas non-Galerkin coarse grids target the same reduced stencil size and associated reduced communication, but with no deterioration in the AMG convergence rate. For example, our parallel tests for a 27-point finite element discretization of the Poisson problem already show the fastest AMG solve phase time is for classical parallel AMG coupled with the non-Galerkin approach. Regardless of which AMG scenario used, the non-Galerkin approach provides a parallel performance benefit.



(a) Best Current Parallel AMG (b) Classical Parallel AMG

Figure 6: Speedup and AMG solve phase timings comparing non-Galerkin AMG and Galerkin AMG on machine with *slow* network.

6 Comments/Conclusions

In conclusion, the proposed non-Galerkin approach is an effective method at reducing coarse grid stencil size and parallel communication, while having a negligible affect on the AMG convergence rate for the problems considered here. In particular, the method relies primarily on row wise computations in parallel, thus making it a relatively inexpensive method with respect to communication. Additionally, as indicated by our experiments, the proposed method is applicable to most AMG codes, include classical RS-AMG and SA.

There are some outstanding issues, including the importance of maintaining symmetry and definiteness on coarse grids. While theoretically attractive, we have not observed maintenance of symmetry being critical to performance, even out to 150 million unknowns in parallel. However, we will maintain this option in our code as we continue experiments on more real world test problems. Maintaining definiteness on coarse grids, outside of the case of M-matrices, will continue to be a future research issue.

References

[1] S. F. Ashby and R. D. Falgout. A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. *Nuclear Science and Engineering*, 124(1):145–159, September 1996. UCRL-JC-122359.

[2] W. N. Bell, L. N. Olson, and J. Schroder. PyAMG: Algebraic multigrid solvers in Python v2.0. `http://www.pyamg.org`, 2011. Release 1.0.

[3] A. Brandt. General highly accurate algebraic coarsening. *Electron. Trans. Numer. Anal.*, 10:1–20, 2000.

[4] A. Brandt, S. F. McCormick, and J. W. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D. J. Evans, editor, *Sparsity and Its Applications*, pages 257–284. Cambridge Univ. Press, Cambridge, 1984.

[5] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial*. SIAM, Philadelphia, PA, USA, 2nd edition, 2000.

[6] H. Gahvari, A. H. Baker, M. Schulz, U. M. Yang, K. E. Jordan, and W. Gropp. Modeling the performance of an algebraic multigrid cycle on HPC platforms. In *Proceedings of the international conference on Supercomputing*, ICS '11, pages 172–181, New York, NY, USA, 2011. ACM.

[7] V. Henson and U. Yang. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41:155–177, 2002.

[8] *hypre*: High performance preconditioners. http://www.llnl.gov/CASC/hypre/.

[9] Y. Notay. Convergence analysis of perturbed two-grid and multigrid methods. *SIAM J. Numer. Anal.*, 45:1035–1044, 2007.

[10] L. N. Olson, J. B. Schroder, and R. S. Tuminaro. A general interpolation strategy for algebraic multigrid using energy-minimization. *SIAM J. Sci. Comput.*, 33:966–991, 2011.

[11] Y. Renard. The GetFem++ Project. `http://download.gna.org/getfem/doc/getfem\textunderscoreproject.pdf`.

[12] J. W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. F. McCormick, editor, *Multigrid Methods*, Frontiers Appl. Math., pages 73–130. SIAM, Philadelphia, 1987.

[13] H. D. Sterck, R. D. Falgout, J. W. Nolting, and U. M. Yang. Distance-two interpolation for parallel algebraic multigrid. *Numer. Linear Algebra Appl.*, 15(2–3):115–139, 2008. Special issue on Multigrid Methods. UCRL-JRNL-230844.

[14] H. D. Sterck, U. Yang, , and J. Heys. Reducing complexity in parallel algebraic multigrid preconditioners. *SIAM J. on Matrix Anal. and Appl.*, 27:1019–1039, 2006.

[15] E. Treister. Sparsified coarsening multigrid for convection diffusion, 2013. Student paper at the Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, March 17-22.

[16] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, London, UK, 2001.

[17] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid based on smoothed aggregation for second and fourth order problems. *Computing*, 56:179–196, 1996.

[18] R. Wienands and I. Yavneh. Collocation coarse approximation (CCA) in multigrid. *SIAM J. Sci. Comput.*, 31:3643–3660, September 2009.